PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : H04L 9/00

A2

(11) International Publication Number: WO 98/47262

(43) International Publication Date: 22 October 1998 (22.10.98)

(21) International Application Number: PCT/US98/07823

(22) International Filing Date: 14 April 1998 (14.04.98)

(30) Priority Data:

60/043,536	14 April 1997 (14.04.97)	US
Not furnished	13 April 1998 (13.04.98)	US

(81) Designated States: BR, CA, CN, JP, KR, MX, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published
Without international search report and to be republished upon receipt of that report.

(71) Applicant: LUCENT TECHNOLOGIES INC. [US/US]; 600 Mountain Avenue, Murray Hill, NJ 07974–0636 (US).

(72) Inventors: ETZEL, Mark, H.; 9 Quarry Lane, Harvard, MA 01451 (US). FRANK, Robert, John; 1200 Gresham Road, Silver Spring, MD 20904 (US). HEER, Daniel, Nelson; 29 Thornell Road, Newton, NH 03858 (US). NcNELIS, Robert, John; 10075 Quantrell Row, Columbia, MD 21046 (US). MIZIKOVSKY, Semyon, B.; 227 Yellowknife Road, Morganville, NJ 07751 (US). RANCE, Robert, John; 6 Wintergreen Circle, Andover, MA 01810 (US). SHIPP, R., Dale; 5351 Hesperus Drive, Columbia, MD 21044 (US).

(74) Agents: GOO, Jimmy et al.; Lucent Technologies Inc., P.O. Box 679, Holmdel, NJ 07733–3030 (US).

(54) Title: METHODS AND APPARATUS FOR MULTIPLE–ITERATION CMEA ENCRYPTION AND DECRYPTION FOR IMPROVED SECURITY FOR WIRELESS TELEPHONE MESSAGES

(57) Abstract

An enhanced CMEA encryption system suitable for use in wireless telephony. A plaintext message is introduced into the system and subjected to a first iteration of a CMEA process, using a first CMEA key to produce an intermediate ciphertext. The intermediate ciphertext is then subjected to a second iteration of the CMEA process using a second CMEA key to produce a final ciphertext. Additional security is achieved by subjecting the plaintext and intermediate ciphertext to input and output transformations before and after each iteration of the CMEA process. The CMEA iterations may be performed using an improved use of a box function which adds permutations to a message or intermediate crypto–processed data. Decryption is achieved by subjecting a ciphertext message to the reverse order of the steps used for encryption, replacing the input and output transformations by inverse output and inverse input transformations, respectively, as appropriate.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	**ES**	Spain	**LS**	Lesotho	**SI**	Slovenia
AM	Armenia	**FI**	Finland	**LT**	Lithuania	**SK**	Slovakia
AT	Austria	**FR**	France	**LU**	Luxembourg	**SN**	Senegal
AU	Australia	**GA**	Gabon	**LV**	Latvia	**SZ**	Swaziland
AZ	Azerbaijan	**GB**	United Kingdom	**MC**	Monaco	**TD**	Chad
BA	Bosnia and Herzegovina	**GE**	Georgia	**MD**	Republic of Moldova	**TG**	Togo
BB	Barbados	**GH**	Ghana	**MG**	Madagascar	**TJ**	Tajikistan
BE	Belgium	**GN**	Guinea	**MK**	The former Yugoslav Republic of Macedonia	**TM**	Turkmenistan
BF	Burkina Faso	**GR**	Greece			**TR**	Turkey
BG	Bulgaria	**HU**	Hungary	**ML**	Mali	**TT**	Trinidad and Tobago
BJ	Benin	**IE**	Ireland	**MN**	Mongolia	**UA**	Ukraine
BR	Brazil	**IL**	Israel	**MR**	Mauritania	**UG**	Uganda
BY	Belarus	**IS**	Iceland	**MW**	Malawi	**US**	United States of America
CA	Canada	**IT**	Italy	**MX**	Mexico	**UZ**	Uzbekistan
CF	Central African Republic	**JP**	Japan	**NE**	Niger	**VN**	Viet Nam
CG	Congo	**KE**	Kenya	**NL**	Netherlands	**YU**	Yugoslavia
CH	Switzerland	**KG**	Kyrgyzstan	**NO**	Norway	**ZW**	Zimbabwe
CI	Côte d'Ivoire	**KP**	Democratic People's Republic of Korea	**NZ**	New Zealand		
CM	Cameroon			**PL**	Poland		
CN	China	**KR**	Republic of Korea	**PT**	Portugal		
CU	Cuba	**KZ**	Kazakstan	**RO**	Romania		
CZ	Czech Republic	**LC**	Saint Lucia	**RU**	Russian Federation		
DE	Germany	**LI**	Liechtenstein	**SD**	Sudan		
DK	Denmark	**LK**	Sri Lanka	**SE**	Sweden		
EE	Estonia	**LR**	Liberia	**SG**	Singapore		

METHODS AND APPARATUS FOR MULTIPLE-ITERATION CMEA ENCRYPTION AND DECRYPTION FOR IMPROVED SECURITY FOR WIRELESS TELEPHONE MESSAGES

This application claims the benefit of United States Provisional application serial No. 60/043,536 filed April 14, 1997. The related application entitled "Methods and Apparatus for Enhanced Security Expansion of a Secret Key into a Lookup Table for Improved Security for Wireless Telephone Messages" and filed on even date herewith is noted and incorporated by reference herein in its entirety.

Field of the Invention

The present invention relates generally to wireless telephone cryptography. More particularly, the invention relates to an improved security cryptosystem for rapid and secure encryption in a wireless telephone system.

Background of the Invention

Wireless telephony uses messaging for several purposes including, for example, conveying status information, reconfiguring operating modes, handling call termination, and conveying system and user data such as a subscriber's electronic serial number and telephone number, as well as conversations and other data transmitted by the user. Unlike ordinary wire telephony, in which a central serving station is connected to each subscriber by wire, thus ensuring a fair degree of protection from eavesdropping and tampering by an unauthorized party (attacker), wireless telephone serving stations (i.e., base stations) must transmit and receive messages via signals over the air, regardless of the physical location of the subscribers.

Because the base station must be able to send and receive messages to and from a subscriber anywhere, the messaging process is wholly dependent on signals received from and sent to the subscriber equipment. Because the signals are transmitted over the air, they can be intercepted by an eavesdropper or interloper with the right equipment.

If a signal is transmitted by a wireless telephone in plaintext, a danger exists that an eavesdropper will intercept the signal and use it to impersonate a subscriber, or to intercept private data transmitted by the user. Such private data may include the content of

conversations. Private data may also include non-voice data transmitted by the user such as, for example, computer data transmitted over a modem connected to the wireless telephone, and may also include bank account or other private user information transmitted typically by means of keypresses. An eavesdropper listening to a conversation or intercepting non-voice data may obtain private information from the user. The message content of an unencrypted telephone signal (i.e., plaintext signal) is relatively easily intercepted by a suitably adapted receiver.

Alternatively, an interloper can interject himself into an established connection by using a greater transmitting power, sending signals to the base station, and impersonating a party to the conversation.

In the absence of applying cryptography to messages being transmitted by wireless signals, unauthorized use of telephone resources, eavesdropping of messages, and impersonation of called or calling parties during a conversation are possible. Such unauthorized interloping and/or eavesdropping has in fact proven to be a grave problem and is highly undesirable.

The application of cryptography to wireless telephone applications offers a solution to the security problems discussed above, but the application of standard cryptography methods to wireless telephony has encountered significant difficulties due to the computationally-intensive nature of these methods. Specifically, these methods are subject to the constraints imposed by the desire to furnish a small wireless handset and the constraints on processing power imposed by the small size of the handset. The processing power present in typical wireless handsets is insufficient to handle the processing requirements of commonly known cryptographic algorithms such as DES (Data Encryption Standard). Implementing such a commonly known cryptographic algorithm in a typical wireless telephone system would potentially increase the time needed to process signals (i.e., encrypt and decrypt), thereby causing unacceptable delays for subscribers.

One cryptographic system for wireless telephony is disclosed in Reeds U.S. Patent 5,159,634 ("Reeds"), incorporated herein by reference. Reeds describes a cryptographic system incorporated in a cryptographic algorithm known as the Cellular Message Encryption Algorithm (CMEA) process. There exists a desire to substantially improve this and other

presently existing cryptographic systems for wireless telephony consistent with the resources available in this context.

Summary of the Invention

The present invention advantageously addresses this and other desires. In one method according to the present invention, first and second CMEA keys are generated. Plaintext is introduced, and subjected to a first input transformation to produce a first input transformed message. The first input transformed message is processed by a first iteration of a CMEA process using the first CMEA key to produce a first intermediate ciphertext. This first intermediate ciphertext is subjected to a first output transformation to produce a first output transformed message. The first output transformed message is subjected to a second input transformation to produce a second input transformed message. The second input transformed message is processed by a second iteration of the CMEA process using the second CMEA key to produce a second intermediate ciphertext. The second intermediate ciphertext is subjected to a second output transformation to produce a second output transformed message. According to another aspect of the present invention, the first and second iterations of the CMEA process employ tbox functions with inputs permuted by secret offsets. According to another aspect of the present invention, the plaintext may be processed by first and second iterations of the CMEA process using first and second CMEA keys, without being subjected to input and output transformations. Encrypted text may be suitably decrypted according to the teachings of the present invention by introducing ciphertext and reversing in order and inverting the steps applied to encrypt plaintext.

An apparatus according to the present invention generates text and supplies it to an I/O interface which identifies it as generated text and supplies the text and the identification to an encryption/decryption processor, which in turn encrypts the text and supplies it to a transceiver for transmission. When the apparatus receives a transmission via the transceiver, the transmission is identified as incoming ciphertext, and the ciphertext and identification are supplied to the encryption/decryption processor which decrypts the ciphertext and supplies it as text to the I/O processor for routing to its destination. In a preferred embodiment, this apparatus is integrated into a wireless phone utilizing a standard microprocessor and memory consistent with those presently typically employed in such phones.

A more complete understanding of the present invention, as well as further features and advantages of the invention, will be apparent from the following Detailed Description and the accompanying drawings.

Brief Description of the Drawings

Fig. 1 is a flowchart illustrating a prior art CMEA key generation process and CMEA implementation;

Fig. 2 is a flowchart illustrating an enhanced CMEA encryption method employing multiple CMEA iterations according to the present invention;

Fig. 3 is a flowchart illustrating an enhanced CMEA encryption method according to the present invention employing multiple CMEA iterations, each iteration being preceded by an input transformation and followed by an output transformation;

Fig. 4 is a detailed illustration of an input transformation suitable for use in an encryption method according to the present invention;

Fig. 5 is a detailed illustration of an output transformation suitable for use in an encryption method according to the present invention;

Fig. 6 is a flowchart illustrating a method according to the present invention of decrypting ciphertext encrypted by an enhanced CMEA process; and

Fig. 7 is a diagram illustrating a telephone set employing enhanced CMEA encryption according to the present invention.

Detailed Description

Fig. 1 is a diagram illustrating a prior art method 100 using a CMEA key for encryption of certain critical user data which may be transmitted during a call. The creation and description of the CMEA key are well known in the art. The CMEA key is used to create a secret array, tbox(z), of 256 bytes. Alternatively, the tbox may be implemented as a function call. This implementation reduces the use of RAM, but increases processing time by roughly an order of magnitude.

At step 102, unprocessed text is introduced. At step 104, in systems which implement tbox as a static table rather than a function call, the static tbox table is derived. The tbox table is derived as follows. For each z in the range $0 \le z < 256$, tbox(z) = C(((C(((C(((C((z XOR k0)+k1)+z)XOR k2)+k3)+z)XOR k4)+k5)+z)XOR k6)+k7)+z, where "+" denotes modulo

256 addition, "XOR" is the bitwise boolean XOR operator, "z" is the function argument, k0,. .,k7 comprise the eight octets of the CMEA key, and C( ) is the outcome of a CAVE 8-bit table look-up.

CMEA comprises three successive stages, each of which alters each byte string in the data buffer. At steps 106, 108 and 110, first, second and third stages of the CMEA process are respectively performed, as will be described herein. A data buffer d bytes long, with each byte designated by b(i), for an integer in the range $0 \leq i < d$, is enciphered in three stages. The first stage (I) of CMEA is as follows:

1. Initialize a variable z to zero,
2. For successive integer values of i in the range $0 \leq i < d$
   a. form a variable q by: q = z (+) low order byte of i, where (+) is the bitwise boolean Exclusive-OR operator,
   b. form variable k by: k = TBOX(q),
   c. update b(i) with: b(i) = b(i) + k mod 256, and
   d. update z with: z = b(i) + z mod 256.

The second stage (II) of CMEA is:

1. for all values of i in the range $0 \leq i < (d - 1)/2$: $b(i) = b(i) \oplus (b(d - 1 - i) \text{ OR } 1)$, where OR is the bitwise boolean OR operator.

The final or third stage (III) of CMEA is the decryption that is inverse of the first stage:

1. Initialize a variable z to zero,
2. For successive integer values of i in the range $0 \leq i < d$
   a. form a variable q by: $q = z \oplus$ low order byte of i,
   b. form variable k by: k = TBOX(q),
   c. update z with: z= b(i) + z mod 256,
   d. update b(i) with b(i) = b(i) - k mod 256.

At step 112, the final processed output is provided.

The CMEA process is self-inverting. That is, the same steps applied in the same order are used both to encrypt plaintext and to decrypt ciphertext. Therefore, there is no need to determine whether encryption or decryption is being carried out. Unfortunately, it has been

shown that the CMEA process is subject to an attack which will allow recovery of the CMEA key used for a call.

In order to provide added security to customer information, an encryption system according to the present invention preferably performs two iterations of the CMEA process, with a different key used in each iteration. A first input transformation and a first output transformation are performed before and after the first iteration of the CMEA process, and a second input transformation and a second output transformation are performed after the second iteration of the CMEA process. An alternative encryption system according to the present invention preferably improves the use of the tbox function by adding at least one permutation of the tbox inputs in one or more iterations of the CMEA process. The improved use of the tbox function is disclosed in our patent application Serial number _____, entitled "Methods and Apparatus for Enhanced Security Expansion of a Secret Key into a Lookup Table for Improved Security for Wireless Telephone Messages" filed on even date with the present application and incorporated herein by reference. In another aspect of the invention, first and second iterations of the CMEA process may be performed, but without the input and output transformations before and after each iteration of the CMEA process.

Fig. 2 is a flowchart showing the steps performed by an encryption process 200 according to an aspect of the present invention. The encryption process of Fig. 2 includes two iterations of the CMEA process described in connection with the discussion of Fig. 1, with a different CMEA key used for each iteration. At step 202, the plaintext is introduced into the encryption process. At step 204, the plaintext is encrypted in a first iteration using the CMEA process, using a first CMEA key. At step 206, the first iteration is completed, and intermediate ciphertext is produced. At step 208, the intermediate ciphertext is subjected to a second iteration of the CMEA process, using a second CMEA key. At step 210, the final ciphertext is produced.

Fig. 3 is a diagram illustrating an encryption process 300 according to another aspect of the present invention. At step 302, the plaintext message is introduced into the encryption process. At step 304, the plaintext message is subjected to a first input transformation to produce a first input transformed message. At step 306, the first input transformed message

is subjected to a first iteration of a CMEA process using a first CMEA key to produce a first intermediate ciphertext. Preferably the first iteration of the CMEA process employs an improved use of the tbox function in which each input of the tbox function is subjected to a permutation. The improved use of the tbox function is disclosed in our aforementioned application Serial No. _____. At step 308, the output of the first iteration of the CMEA process is subjected to a first output transformation to produce a first output transformed message. At step 310, the first intermediate ciphertext is subjected to a second input transformation to produce a second input transformed message. At step 312, the transformed intermediate ciphertext is subjected to a second iteration of the CMEA process, using a second CMEA key to produce a second intermediate ciphertext. The second iteration of the CMEA process preferably employs the improved use of the tbox function described in our above mentioned application. At step 314, the second intermediate ciphertext is subject to a second output transformation to produce a second output transformed message. At step 316, second output transformed message is output as final ciphertext.

Fig. 4 is a diagram illustrating in detail an input transformation 400 which may suitably be used in the encryption process 300 described in connection with Fig. 3. The inverse input transformation 400 is self-inverting. Each of j+1 input data octets, j+1, j, . . ., 2, 1 is XORed with a transformation octet. The transformation octet is a secret value which may be created using any of a number of techniques commonly used in the art. Two transformation octets are preferably used and applied in alternating fashion to the input data octets. Transformation octet I_2 is applied to input data octet j+1, transformation octet I_1 is applied to input data octet j, transformation octet I_2 is applied to input data octet j-1, and so on. The application of the transformation produces a new set of input data octets j+1', j', . . ., 2', 1', which is then used as described above in connection with the discussion of Fig. 3.

Fig. 5 is a diagram illustrating a forward/inverse output transformation 500 which may suitably be used in the encryption process 300 described in connection with Fig. 3. For the forward output transformation, each of j+1 output data octets, j+1, j, . . ., 2, 1 is summed with a transformation octet. The transformation octet is a secret value which may be created using any of a number of techniques commonly used in the art. For the inverse output transformation, the summation is replaced with a subtraction. Two transformation octets are

preferably used and applied in alternating fashion to the output data octets. Transformation octet O_2 is applied to output data octet j+1, transformation octet O_1 is applied to output data octet j, transformation octet O_2 is applied to output data octet j-1, and so on. The application of the transformation produces a new set of output data octets j+1', j', . . ., 2', 1', which is then used as described above in connection with the discussion of Fig. 3.

Because the encryption system of the present invention requires the application of two keys, it is not self-inverting. That is, the same operations applied in the same order will not either encrypt plaintext or decrypt ciphertext. Moreover, the output transformation described in connection with the discussion of Fig. 5 is not self-inverting. Therefore, a separate decryption process is necessary, as described below.

Fig. 6 illustrates a decryption process 600 according to an aspect of the present invention. Essentially, the steps illustrated in Fig. 3 are followed, but in the reverse of the order shown in Fig. 3. First and second inverse input and output transformations are employed in place of the input and output transformations of Fig. 3. The first inverse input transformation is simply the second input transformation described above in connection with the discussion of Fig. 3, and the second inverse input transformation is the first input transformation described above in connection with the discussion of Fig. 3.

At step 602, the ciphertext message is introduced into the decryption process. At step 604, the ciphertext message is subjected to a first inverse output transformation to produce a first inverse output transformed message. The first inverse output transformation is the inverse of the second output transformation described in connection with Fig. 3 and in more detail in connection with Fig. 5. In particular, the addition step in the output transformation is canceled by a subtraction in the inverse output transformation. At step 606, the first inverse output transformed message is subjected to a first iteration of the CMEA process to produce a first intermediate decrypted ciphertext message. The first iteration of the CMEA process preferably employs an improved use of the tbox function according to our aforementioned application Serial No. _____. The keying used for this first iteration is the second CMEA key and the second tbox input permutation. At step 608, the first intermediate ciphertext is subjected to a first inverse input transformation, which is identical to the second input transformation described in connection with the discussion of Fig. 3, to produce a first

inverse input transformed message. Next, at step 610, the first inverse input transformed message is subjected to a second inverse output transformation, which is the inverse of the first output transformation described in connection with the discussion of Fig. 3, to produce a second inverse output transformed message. At step 612, the second inverse output transformed message is subjected to a second iteration of the CMEA process to produce a second intermediate decrypted ciphertext message. The second iteration of the CMEA process preferably employs the improved use of the tbox function. The keying used for this iteration of the modified CMEA process is the first CMEA key and the first tbox input permutation. At step 616, the second intermediate decrypted ciphertext message is subjected to a second inverse input transformation, which is identical to the first input transformation described in connection with the discussion of Fig. 4 to produce a second inverse input transformed message. At step 618, the second iteration is completed and the second inverse input transformed message is output as the final plaintext.

The encryption described in connection with the discussion of Fig. 2 can be similarly reversed. In order to decrypt a message encrypted according to the aspect of the invention described in connection with Fig. 2 above, the decryption described at Fig. 6 is executed, but without executing the inverse input and output transformations.

Because the decryption described in connection with Fig. 6 cannot be accomplished simply by operating the encryption described in connection with Fig. 3, it is necessary for a device using encryption and decryption systems according to the present invention to recognize when a message needs to be encrypted and when it needs to be decrypted.

Fig. 7 is a diagram showing a wireless telephone set 700 equipped to perform message transmission and encryption/decryption according to the present invention, with facilities both for recognizing whether a message needs to be encrypted or decrypted, and for performing the appropriate encryption or decryption. The telephone set 700 includes a transceiver 702, an input/output (I/O) interface 704, an encryption/decryption processor 706, and a key generator 708. The key generator 708 receives and employs stored secret data for key generation. Stored secret data is preferably stored in nonvolatile memory 710 such as an EEPROM or a Flash memory. The key generator 708 stores the generated keys in memory 712. The encryption/decryption processor also includes memory 714 for storage of keys

received from the key generator 708, a static tbox table which may be generated and used if it is desired to implement the tbox function as a static table, and other values which may be produced and stored during encryption and decryption. The telephone set 700 also includes a message generator 716, which generates messages to be encrypted by the encryption/decryption processor 706 and transmitted by the transceiver 702.

When an internally generated message is to be encrypted and transmitted by the telephone set 700, the message is transmitted from message generator 712 to the I/O interface 704. The I/O interface 704 identifies the message as an internally generated message to be encrypted and transmits the message, along with the identification, to the encryption/decryption processor 706. The encryption/decryption processor 706 receives one or more keys from the key generator 708, which it then uses to encrypt the message. Preferably, the encryption decryption processor 706 receives two keys from the key generator 708, which are then employed to perform two-iteration CMEA encryption using an input and output transformations as described above in connection with Fig. 3.

The encryption/decryption processor 706 subjects the plaintext message to a first input transformation to produce a first input transformed message. Next, the first input transformed message is subjected to a first iteration of a CMEA process using a first CMEA key, to produce a first intermediate ciphertext message. The first iteration of the CMEA process may suitably employ the improved use of the tbox function in which each tbox function input is subjected to a permutation. The first intermediate ciphertext message is subjected to a first output transformation to produce a first output transformed message. Next, the first output transformed message is subjected to a second input transformation to produce a second input transformed message. The second input transformed message is then subjected to a second iteration of the modified CMEA process, using a second CMEA key, to produce a second intermediate ciphertext message. The second iteration process may also suitably employ the improved use of the tbox function. The output of the second iteration of the CMEA process is then subjected to a second output transformation to produce a second output transformed message. Finally, the second iteration is completed and the second output transformed message is produced as the final ciphertext. Upon completion of the encryption,

the final ciphertext may be stored in memory 714, and is routed to the I/O interface 704 and to the transceiver 702 for transmission.

When an encrypted message is received by the telephone set 700, the transceiver 702 passes it to the I/O interface 704. The I/O interface identifies the message as an encrypted message, and passes this identification, along with the message, to the encryption/decryption processor 706. The encryption/decryption processor 706 receives one or more keys from the key generator 708 and decrypts the message, preferably using a two-iteration CMEA decryption process as described in connection with Fig. 6.

When the encryption/decryption processor 706 receives a ciphertext message from the I/O interface 704, the ciphertext message is subjected to a first inverse output transformation to produce a first inverse output transformed message. The first inverse output transformation is the inverse of the second output transformation described in connection with Fig. 3 and in more detail in connection with Fig. 5. In particular, the addition step in the output transformation is canceled by a subtraction in the inverse output transformation. Next, a first iteration of the CMEA process is performed, preferably employing the improved use of the tbox function to produce a first intermediate decrypted ciphertext message. The keying used for this first iteration is the second CMEA key and the second tbox input permutation. Next, the first intermediate decrypted ciphertext message is subjected to a first inverse input transformation to produce a first inverse input transformed message. The first inverse input transformation is identical to the second input transformation described in connection with the discussion of Fig. 3. The second inverse input transformed message is then subjected to a second inverse output transformation to produce a second inverse output transformed message. The second inverse output transformation is the inverse of the first output transformation described in connection with the discussion of Fig. 3. A second iteration of the CMEA process is then performed, preferably employing the improved use of the tbox function, to produce a second intermediate decrypted ciphertext message. The keying used for this iteration of the modified CMEA process is the first CMEA key and the first tbox permutation. The second intermediate decrypted ciphertext message is then subjected to second inverse input transformation to produce a second inverse input transformed message. The second inverse input transformation is identical to the first input transformation described

in connection with the discussion of Fig. 3. The second inverse input transformed message is passed as plaintext to the I/O interface 704, where it is then routed for its ultimate use.

The above-described enhancements to the CMEA process, while substantially increasing security, do not substantially increase processing or system resources, and are therefore well suited to use in an environment such as a wireless telephone system, in which units such as the mobile units often have limited processing power.

While the present invention is disclosed in the context of a presently preferred embodiment, it will be recognized that a wide variety of implementations may be employed by persons of ordinary skill in the art consistent with the above discussion and the claims which follow below.

performing a second iteration of the CMEA process on the second input transformed message, using a second CMEA key, to produce a second intermediate ciphertext message; and

performing a second output transformation on the second intermediate ciphertext message to produce a second output transformed message.

7. The method of claim 6 wherein the first input transformed message is subjected to a tbox function including permutation of each input to a tbox function by first and second secret offsets during the first iteration of the CMEA process and the second input transformed message is subjected to a tbox function including permutation of each input to the tbox function by third and fourth secret offsets during the second iteration of the CMEA process.

8. A method of decryption of a ciphertext message, comprising the steps of:

introducing a ciphertext message;

performing a first inverse output transformation on the ciphertext message to produce a first inverse output transformed message;

performing a first iteration of a CMEA process on the first inverse output transformed message, using a second CMEA key, to produce a first intermediate decrypted ciphertext message; and

performing a first inverse input transformation on the first intermediate decrypted ciphertext message to produce a first inverse input transformed message.

9. The method of claim 8 and also including the steps of:

performing a second inverse output transformation on the first inverse input transformed message to produce a second inverse output transformed message;

performing a second iteration of the CMEA process on the second inverse output transformed message, using a first CMEA key, to produce a second intermediate decrypted ciphertext message; and

performing a second inverse input transformation on the second intermediate decrypted ciphertext message to produce a second inverse input transformed message.

10. The method of claim 9 and also including the steps of retrieving first, second, third and fourth offsets created during encryption of a plaintext message to produce the

a key generator for receiving secret data from the memory and using the secret data to generate one or more encryption keys;

an encryption/decryption processor, the encryption decryption processor receiving a message and a message identification signal from the input/output interface and first and second encryption keys from the key generator, the encryption/decryption processor being operative for each incoming message to:

perform a first input transformation on the message to produce a first input transformed message;

perform a first iteration of a CMEA process on the first input transformed message, using the first encryption key, to produce an intermediate ciphertext message;

perform a first output transformation on the first intermediate ciphertext message to produce a first output transformed message;

perform a second input transformation on the first output transformed message to produce a second input transformed message;

perform a second iteration of the CMEA process on the second input transformed message, using the second encryption key, to produce a second intermediate ciphertext message;

perform a second output transformation on the second intermediate ciphertext message; and

pass the final second intermediate ciphertext message to the input/output interface for appropriate routing.

15. The encrypting telephone of claim 14, wherein the first input transformed message is subjected to a tbox function including permutation of each tbox function input by a first and a second offset during the first iteration of the CMEA process and the second input transformed message is subjected to a tbox function including permutation of each tbox function input by a third and a fourth offset during the second iteration of the CMEA process.

16. The encrypting telephone of claim 14, wherein the encryption/decryption processor is operative for incoming messages to:

perform a first inverse output transformation on the message to produce a first inverse output transformed message;

perform a first iteration of a CMEA process on the message, using the second encryption key, to produce a first decrypted intermediate ciphertext message;

perform a first inverse input transformation on the first decrypted intermediate ciphertext message;

perform a second inverse output transformation on the first decrypted intermediate ciphertext message to produce a second inverse output transformed message;

perform a second iteration of the CMEA process on the second inverse output transformed message, using the first encryption key, to produce a second decrypted intermediate ciphertext message;

perform a second inverse input transformation on the second decrypted intermediate ciphertext message; and

pass the final output message to the input/output interface for appropriate routing.

17. The encrypting telephone of claim 16, wherein the first inverse output transformed message is subjected to a tbox function including permutation of each tbox function input by the third and fourth offsets during the first iteration of the CMEA process and the second inverse output transformed message is subjected to a tbox function including permutation of each tbox function input by the first and the second offsets during the second iteration of the CMEA process.

18. A method of data decryption of a ciphertext message previously encrypted by multiple iterations of a plaintext message, comprising the steps of:

introducing the ciphertext message,

performing a first iteration of a CMEA process on the ciphertext to produce an inter mediate decrypted ciphertext; and

performing one or more further iterations of the CMEA process on the intermediate decrypted ciphertext message, each further iteration of the CMEA process prior to the final iteration producing a further intermediate decrypted ciphertext message, the final iteration of the CMEA process producing a final output plaintext message.

19. The method according to claim 18 wherein each iteration of the CMEA process is performed using a different CMEA key.

20. The method according to claim 19 wherein the iterations of the CMEA process are applied in a reverse order from an order in which the iterations of the CMEA process were previously applied to perform data encryption, and wherein each iteration of the CMEA process employs a CMEA key corresponding to the CMEA key previously employed by a corresponding iteration of the CMEA process to perform data encryption on a previous plaintext message to produce the ciphertext message.

21. The method according to claim 20 wherein each message input to an iteration of the CMEA process is subjected to an inverse output transformation before the iteration of the CMEA process and each message output from an iteration of the CMEA process is subjected to an inverse input transformation following the iteration of the CMEA process, each inverse output transformation being an inverse of an output transformation performed after a corresponding iteration of the CMEA process previously used to perform data encryption, and each inverse input transformation being identical to an input transformation performed before a corresponding iteration of the CMEA process previously used to perform data encryption on a previous plaintext message to produce the ciphertext message.

22. The method according to claim 21 wherein each iteration of the CMEA process employs a tbox function including permutation of each tbox function input by one or more secret offsets.

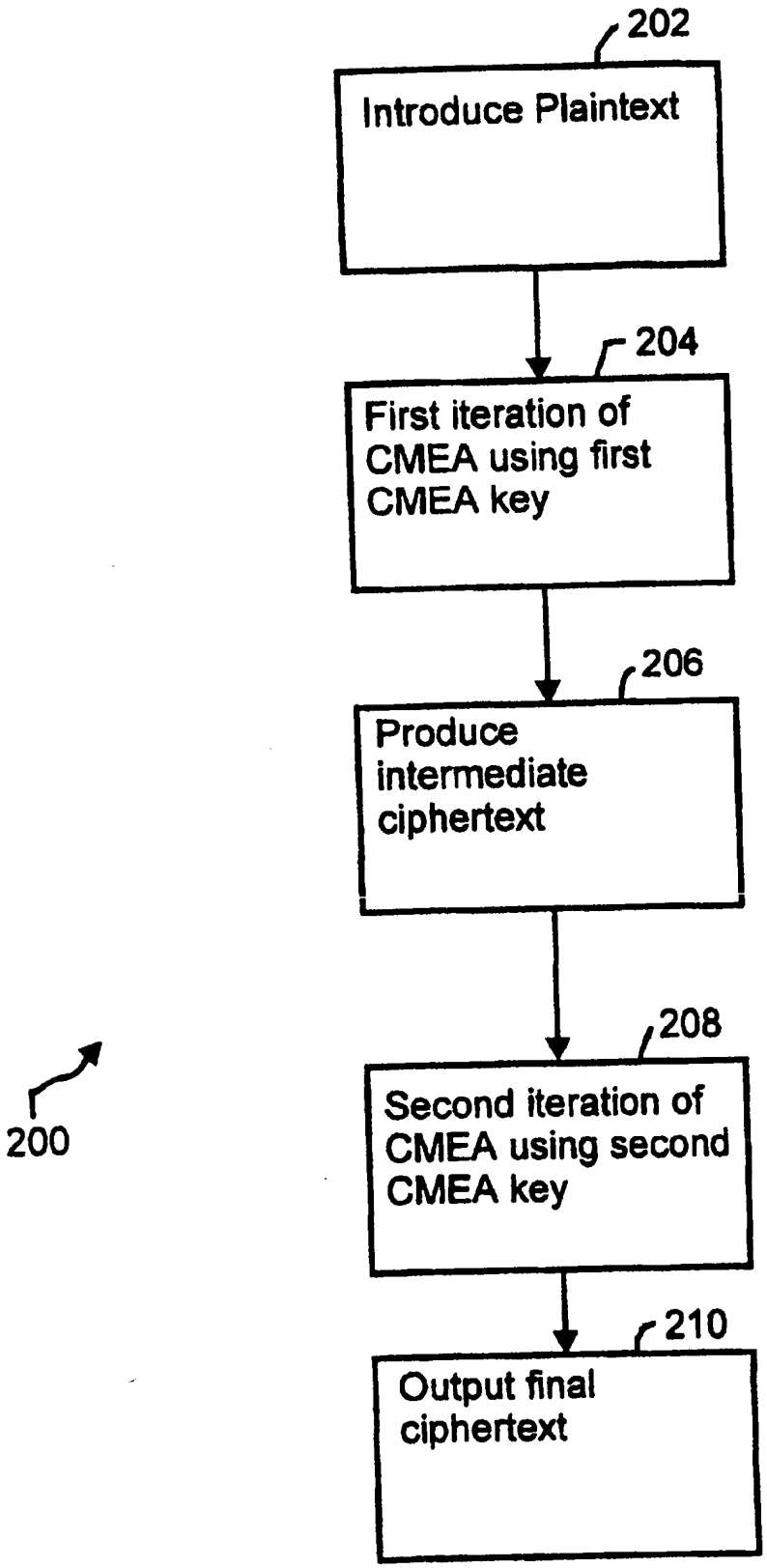
202
Introduce Plaintext
204
First iteration of CMEA using first CMEA key
206
Produce intermediate ciphertext
208
Second iteration of CMEA using second CMEA key
210
Output final ciphertext
200

Fig. 2

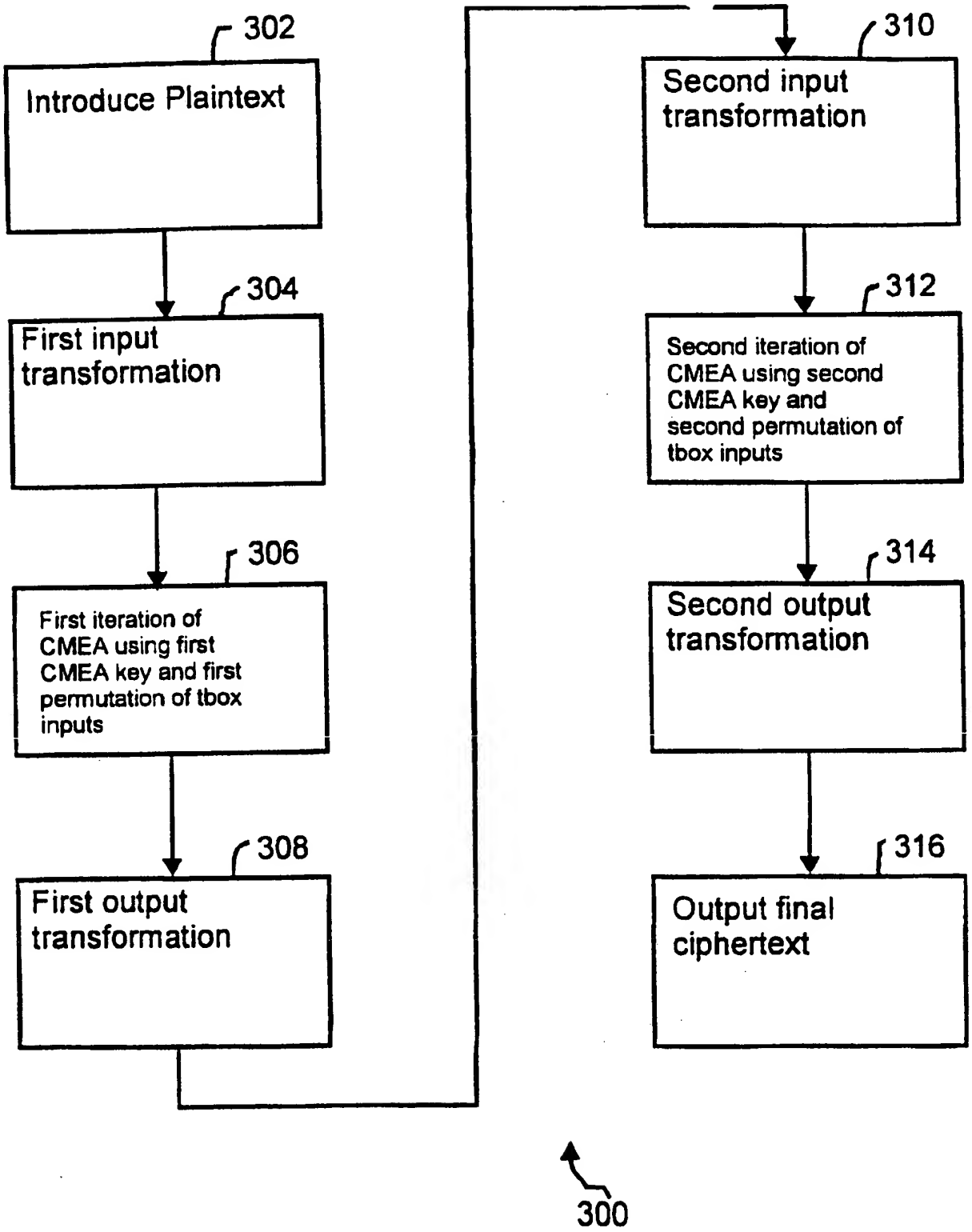
302
Introduce Plaintext
304
First input transformation
306
First iteration of CMEA using first CMEA key and first permutation of tbox inputs
308
First output transformation
310
Second input transformation
312
Second iteration of CMEA using second CMEA key and second permutation of tbox inputs
314
Second output transformation
316
Output final ciphertext
300

Fig. 3

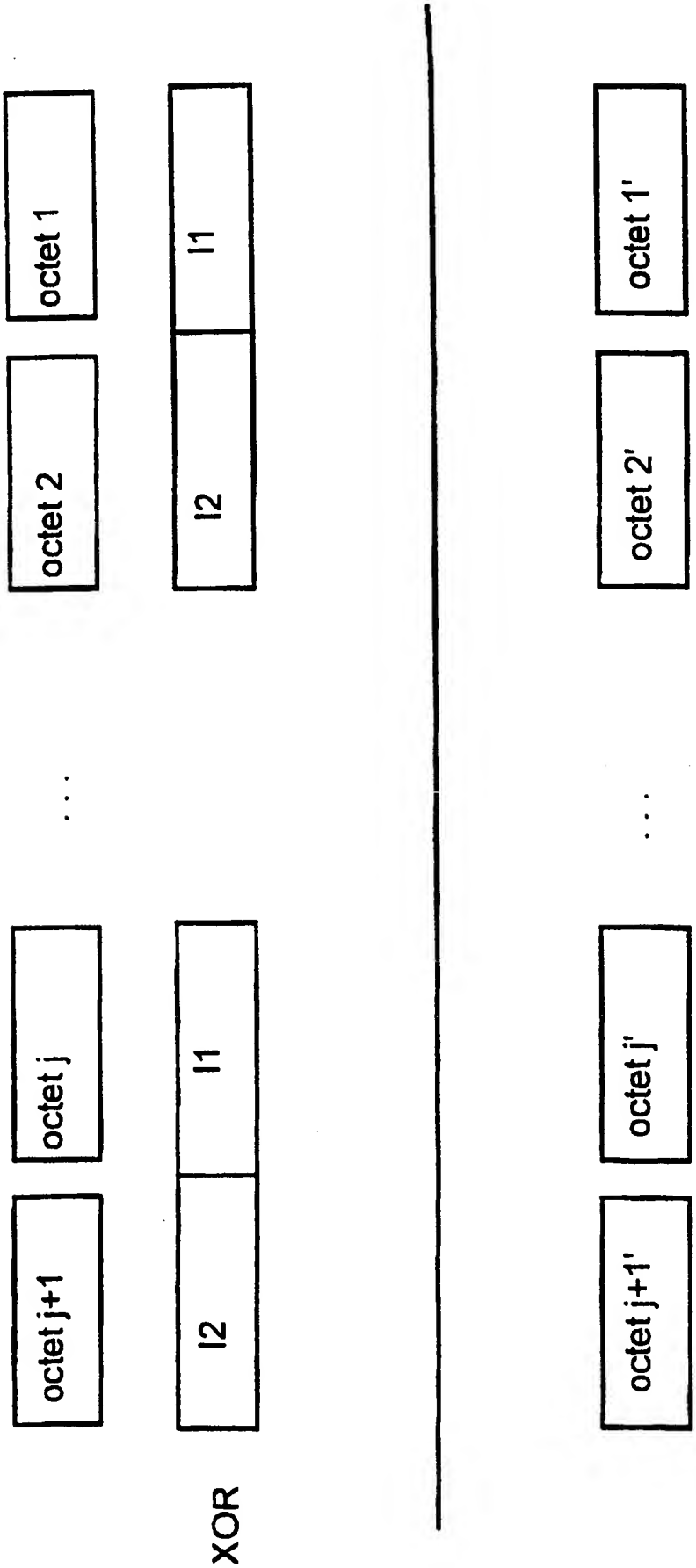
octet 1
octet 2
I1
I2
octet 1'
octet 2'
octet j
octet j+1
I1
I2
octet j'
octet j+1'
XOR
400

Fig. 4

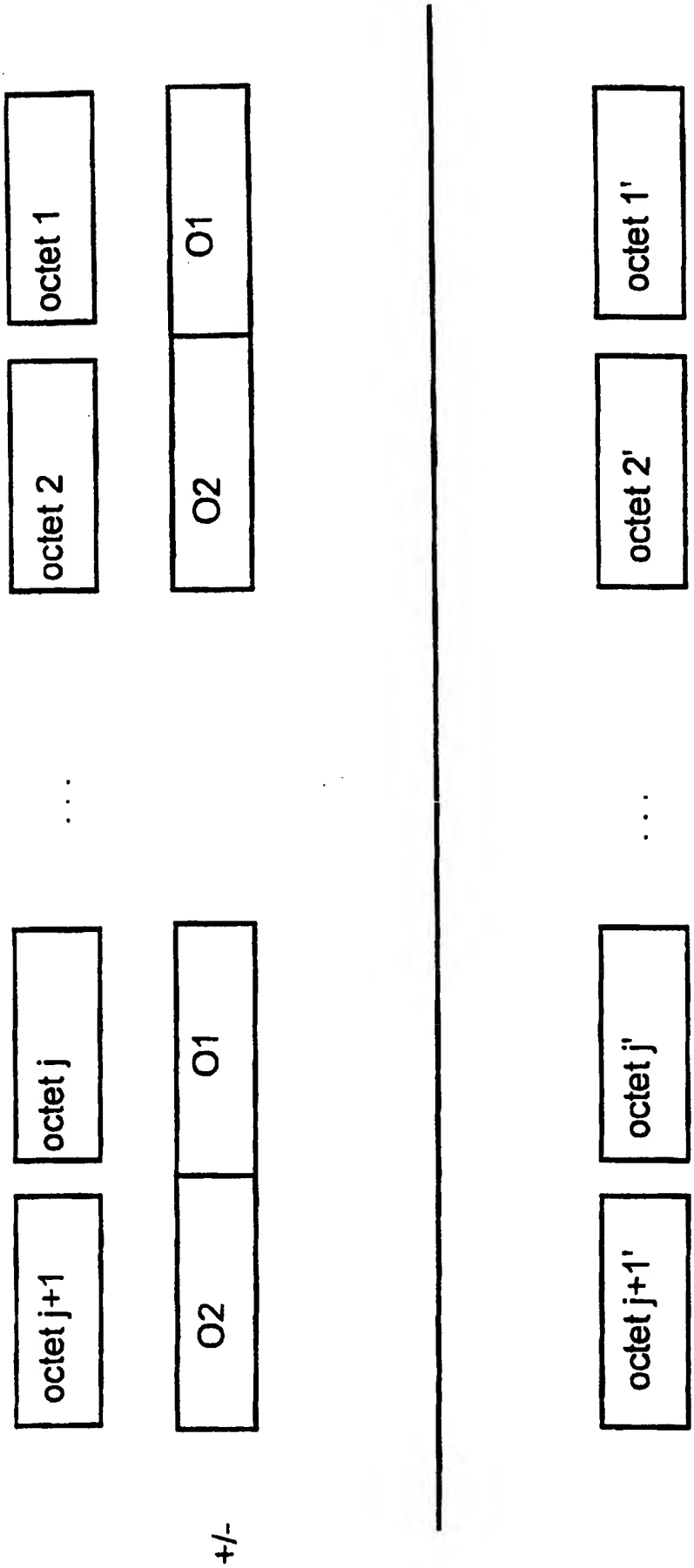
octet 1
octet 2
O1
O2
octet j
octet j+1
O1
O2
+/-
octet 1'
octet 2'
octet j'
octet j+1'
500

Fig. 5

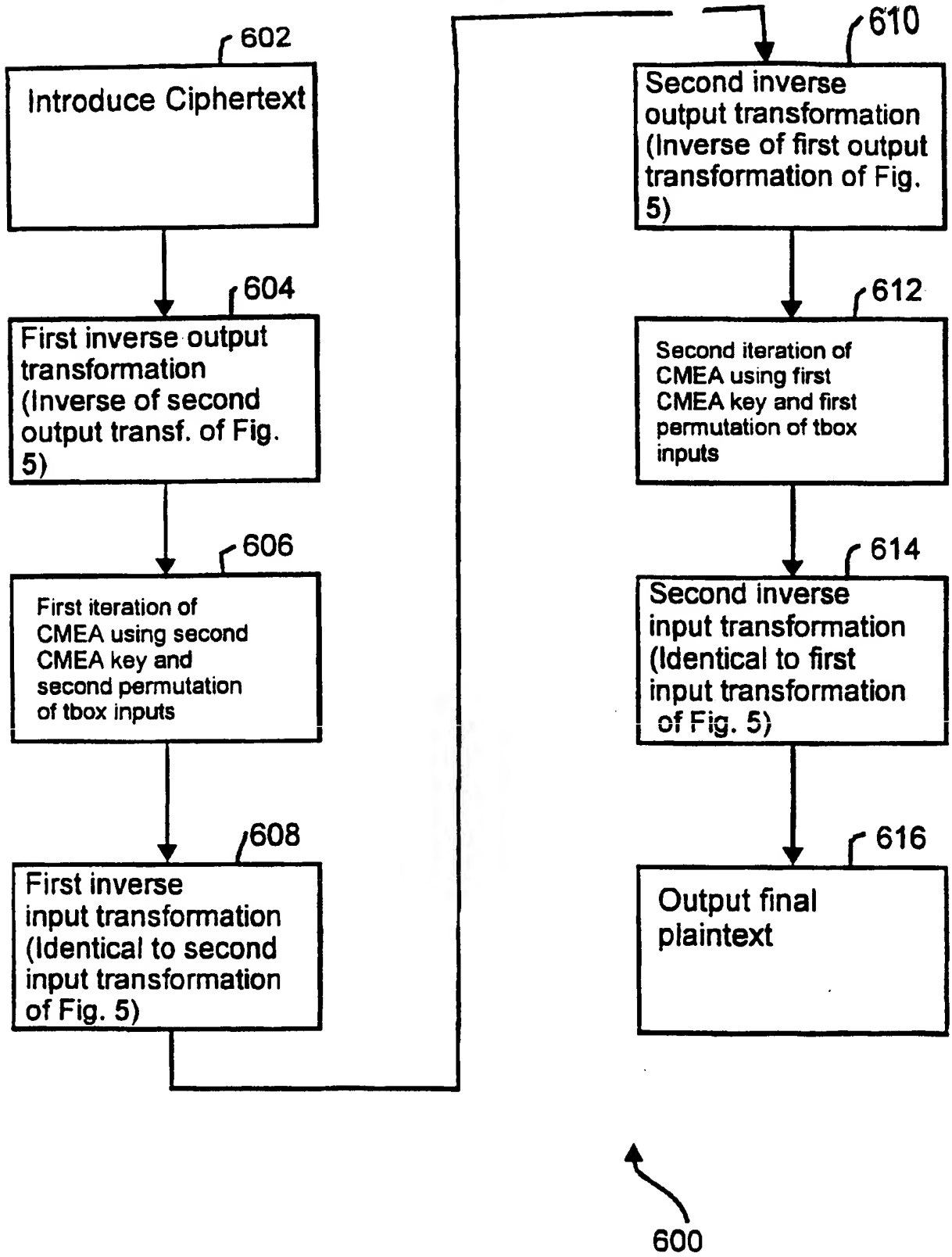
602
Introduce Ciphertext
604
First inverse output transformation (Inverse of second output transf. of Fig. 5)
606
First iteration of CMEA using second CMEA key and second permutation of tbox inputs
608
First inverse input transformation (Identical to second input transformation of Fig. 5)
610
Second inverse output transformation (Inverse of first output transformation of Fig. 5)
612
Second iteration of CMEA using first CMEA key and first permutation of tbox inputs
614
Second inverse input transformation (Identical to first input transformation of Fig. 5)
616
Output final plaintext
600

Fig. 6

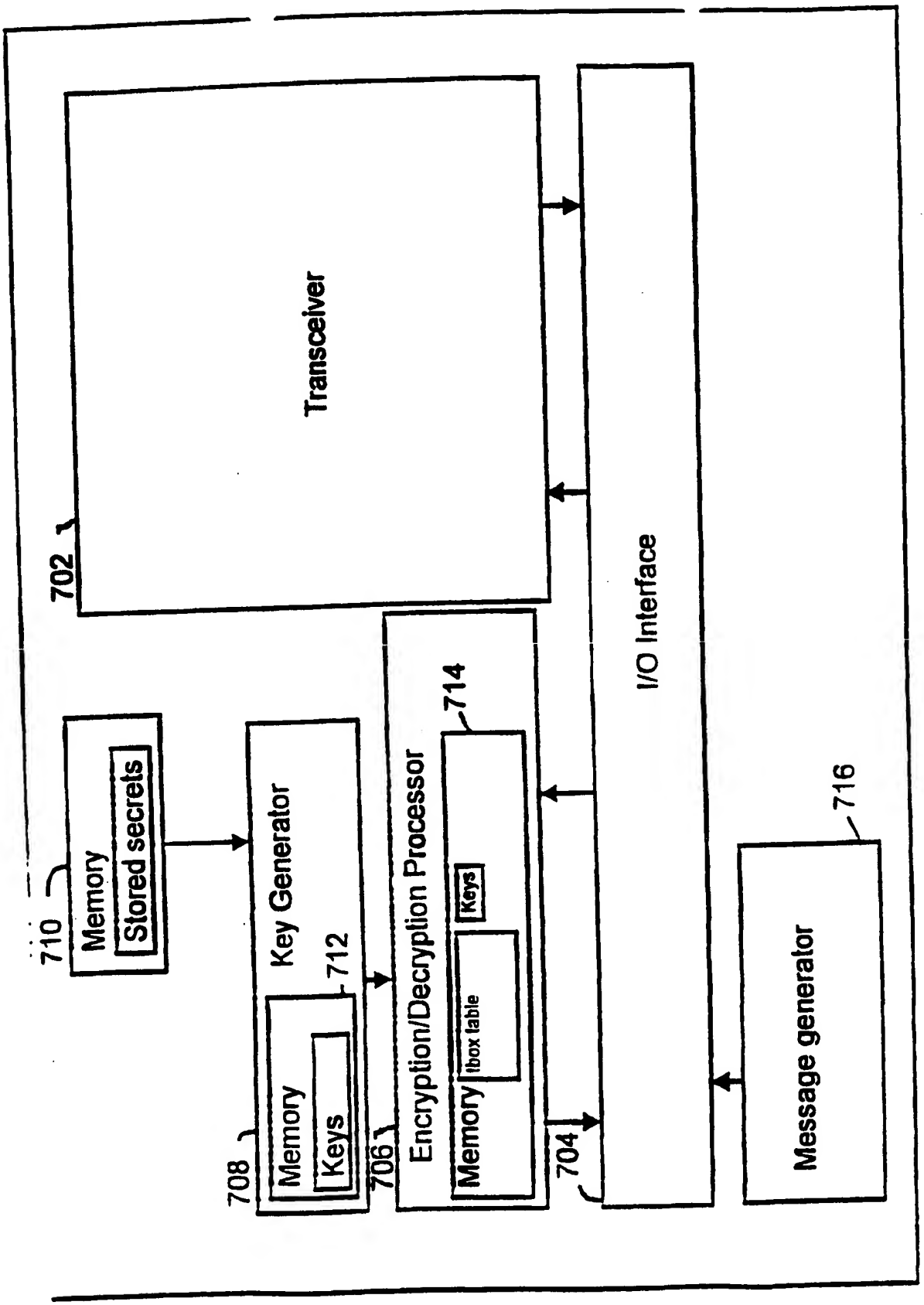
Transceiver
702
I/O Interface
Memory
Stored secrets
710
Key Generator
712
Memory
Keys
708
706
Encryption/Decryption Processor
714
Memory
Ibox lable
Keys
704
Message generator
716
700

Fig. 7